To design a database for IMDb, you can create the following tables/entities to represent the relationships and attributes you've mentioned:

1. Movie Table:

- `movie\_id` (Primary Key)

- `title`

- `release\_date`

- Other movie-related attributes

2. Media Table:

- `media\_id` (Primary Key)

- `movie\_id` (Foreign Key referencing Movie Table)

- `media\_type` (e.g., Video or Image)

- `url` (to store the link to the media file)

3. Genre Table:

- `genre\_id` (Primary Key)

- `genre\_name`

4. MovieGenre Table (for Many-to-Many relationship between Movie and Genre):

- `movie\_id` (Foreign Key referencing Movie Table)

- `genre\_id` (Foreign Key referencing Genre Table)

5. Review Table:

- `review\_id` (Primary Key)

- `movie\_id` (Foreign Key referencing Movie Table)

- `user\_id` (Foreign Key referencing User Table)

- `rating`

- `comment`

- `date`

6. User Table:

- `user\_id` (Primary Key)

- `username`

- `email`

- Other user-related attributes

7. Artist Table:

- `artist\_id` (Primary Key)

- `name`

- Other artist-related attributes

8. Skill Table:

- `skill\_id` (Primary Key)

- `skill\_name`

9. ArtistSkill Table (for Many-to-Many relationship between Artist and Skill):

- `artist\_id` (Foreign Key referencing Artist Table)

- `skill\_id` (Foreign Key referencing Skill Table)

10. Role Table:

- `role\_id` (Primary Key)

- `role\_name`

11. ArtistRole Table (for Many-to-Many relationship between Artist and Role in a Movie):

- `artist\_id` (Foreign Key referencing Artist Table)

- `movie\_id` (Foreign Key referencing Movie Table)

- `role\_id` (Foreign Key referencing Role Table)

-- Create IMDb database

CREATE DATABASE IMDb;

-- Use IMDb database

USE IMDb;

-- Character set

-- want to be able to distinguish text with accents

ALTER DATABASE IMDb CHARACTER SET utf8mb4 COLLATE utf8mb4\_bin;

-- Create tables only

CREATE TABLE Titles (

title\_id VARCHAR(255) NOT NULL, -- not null bc PK

title\_type VARCHAR(50),

primary\_title TEXT, -- some are really long

original\_title TEXT, -- some are really long

is\_adult BOOLEAN,

start\_year INTEGER,

end\_year INTEGER,

runtime\_minutes INTEGER

);

CREATE TABLE Title\_ratings (

title\_id VARCHAR(255) NOT NULL, -- not null bc PK

average\_rating FLOAT,

num\_votes INTEGER

);

CREATE TABLE Aliases (

title\_id VARCHAR(255) NOT NULL, -- not null bc PK

ordering INTEGER NOT NULL, -- not null bc PK

title TEXT NOT NULL,

region CHAR(4),

language CHAR(4),

is\_original\_title BOOLEAN

);

CREATE TABLE Alias\_types (

title\_id VARCHAR(255) NOT NULL, -- not null bc PK

ordering INTEGER NOT NULL, -- not null bc PK

type VARCHAR(255) NOT NULL-- Only stored if not null

);

CREATE TABLE ALias\_attributes (

title\_id VARCHAR(255) NOT NULL, -- not null bc PK

ordering INTEGER NOT NULL, -- not null bc PK

attribute VARCHAR(255) NOT NULL -- only stored if not null

);

CREATE TABLE Episode\_belongs\_to (

episode\_title\_id VARCHAR(255) NOT NULL, -- not null bc PK

parent\_tv\_show\_title\_id VARCHAR(255) NOT NULL,

season\_number INTEGER,

episode\_number INTEGER

);

CREATE TABLE Title\_genres (

title\_id VARCHAR(255) NOT NULL, -- not null bc PK

genre VARCHAR(255) NOT NULL -- not null bc PK

);

-- Names and name is a reserved word in MySQL, so we add an underscore

CREATE TABLE Names\_ (

name\_id VARCHAR(255) NOT NULL, -- not null bc PK

name\_ VARCHAR(255) NOT NULL, -- everybody has a name

birth\_year SMALLINT,

death\_year SMALLINT

);

CREATE TABLE Name\_worked\_as (

name\_id VARCHAR(255) NOT NULL, -- not null bc PK

profession VARCHAR(255) NOT NULL -- not null bc PK

);

-- NOTE: All 3 must must be used as the primary key

-- role is a reserved word in MySQL, so we add an underscore

CREATE TABLE Had\_role (

title\_id VARCHAR(255) NOT NULL, -- not null bc PK

name\_id VARCHAR(255) NOT NULL, -- not null bc PK

role\_ TEXT NOT NULL -- not null bc PK

);

CREATE TABLE Known\_for (

name\_id VARCHAR(255) NOT NULL, -- not null bc PK

title\_id VARCHAR(255) NOT NULL -- not null bc PK

);

CREATE TABLE Directors (

title\_id VARCHAR(255) NOT NULL, -- not null bc PK

name\_id VARCHAR(255) NOT NULL -- not null bc PK

);

CREATE TABLE Writers (

title\_id VARCHAR(255) NOT NULL, -- not null bc PK

name\_id VARCHAR(255) NOT NULL -- not null bc PK

);

CREATE TABLE Principals (

title\_id VARCHAR(255) NOT NULL, -- not null bc PK

ordering TINYINT NOT NULL, -- not null bc PK

name\_id VARCHAR(255) NOT NULL,

job\_category VARCHAR(255),

job TEXT

);

-- SHOW VARIABLES LIKE "local\_infile";

SET GLOBAL local\_infile = 1;

-- Load Aliases.tsv into Aliases table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Aliases.tsv'

INTO TABLE Aliases

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Alias\_attributes.tsv into Alias\_attributes table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Alias\_attributes.tsv'

INTO TABLE Alias\_attributes

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Alias\_types.tsv into Alias\_types table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Alias\_types.tsv'

INTO TABLE Alias\_types

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Directors.tsv into Directors table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Directors.tsv'

INTO TABLE Directors

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Writers.tsv into Writers table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Writers.tsv'

INTO TABLE Writers

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Episode\_belongs\_to.tsv into Episode\_belongs\_to table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Episode\_belongs\_to.tsv'

INTO TABLE Episode\_belongs\_to

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Names\_.tsv into Names\_ table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Names\_.tsv'

INTO TABLE Names\_

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Name\_worked\_as.tsv into Name\_worked\_as table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Name\_worked\_as.tsv'

INTO TABLE Name\_worked\_as

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Known\_for.tsv into Known\_for table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Known\_for.tsv'

INTO TABLE Known\_for

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Principals.tsv into Principals table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Principals.tsv'

INTO TABLE Principals

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Had\_role.tsv into Had\_role table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Had\_role.tsv'

INTO TABLE Had\_role

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Titles.tsv into Titles table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Titles.tsv'

INTO TABLE Titles

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Title\_genres.tsv into Title\_genres table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Title\_genres.tsv'

INTO TABLE Title\_genres

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Load Title\_ratings.tsv into Title\_ratings table

LOAD DATA LOCAL INFILE '/Users/lappy/Git\_repos\_mine/MySQL\_IMDb\_Project/Title\_ratings.tsv'

INTO TABLE Title\_ratings

COLUMNS TERMINATED BY '\t'

IGNORE 1 LINES;

-- Add constraints individually

ALTER TABLE Names\_

ADD CONSTRAINT Names\_pri\_key PRIMARY KEY (name\_id);

ALTER TABLE Titles

ADD CONSTRAINT Titles\_pri\_key PRIMARY KEY (title\_id);

ALTER TABLE Aliases

ADD CONSTRAINT Aliases\_pri\_key PRIMARY KEY (title\_id,ordering);

ALTER TABLE Alias\_attributes

ADD CONSTRAINT Alias\_attributes\_pri\_key PRIMARY KEY (title\_id,ordering);

ALTER TABLE Alias\_types

ADD CONSTRAINT Alias\_types\_pri\_key PRIMARY KEY (title\_id,ordering);

ALTER TABLE Directors

ADD CONSTRAINT Directors\_pri\_key PRIMARY KEY (title\_id,name\_id);

ALTER TABLE Directors

ADD CONSTRAINT Directors\_title\_id\_fkey FOREIGN KEY (title\_id) REFERENCES Titles(title\_id);

ALTER TABLE Writers

ADD CONSTRAINT Writers\_pri\_key PRIMARY KEY (title\_id,name\_id);

ALTER TABLE Writers

ADD CONSTRAINT Writers\_title\_id\_fkey FOREIGN KEY (title\_id) REFERENCES Titles(title\_id);

ALTER TABLE Episode\_belongs\_to

ADD CONSTRAINT Episode\_belongs\_to\_pri\_key PRIMARY KEY (episode\_title\_id);

ALTER TABLE Episode\_belongs\_to

ADD CONSTRAINT Episode\_belongs\_to\_ep\_title\_id\_fkey FOREIGN KEY (episode\_title\_id) REFERENCES Titles(title\_id);

ALTER TABLE Name\_worked\_as

ADD CONSTRAINT Name\_worked\_as\_pri\_key PRIMARY KEY (name\_id,profession);

ALTER TABLE Name\_worked\_as

ADD CONSTRAINT Name\_worked\_as\_name\_id\_fkey FOREIGN KEY (name\_id) REFERENCES Names\_(name\_id);

ALTER TABLE Known\_for

ADD CONSTRAINT Known\_for\_pri\_key PRIMARY KEY (name\_id,title\_id);

ALTER TABLE Known\_for

ADD CONSTRAINT Known\_for\_name\_id\_fkey FOREIGN KEY (name\_id) REFERENCES Names\_(name\_id);

ALTER TABLE Principals

ADD CONSTRAINT Principals\_pri\_key PRIMARY KEY (title\_id,ordering);

-- role\_ is TEXT, so we need to add indexing length (255)

ALTER TABLE Had\_role

ADD CONSTRAINT Had\_role\_pri\_key PRIMARY KEY (title\_id,name\_id,role\_(255));

ALTER TABLE Title\_genres

ADD CONSTRAINT Title\_genres\_pri\_key PRIMARY KEY (title\_id,genre);

ALTER TABLE Title\_genres

ADD CONSTRAINT Title\_genres\_title\_id\_fkey FOREIGN KEY (title\_id) REFERENCES Titles(title\_id);

ALTER TABLE Title\_ratings

ADD CONSTRAINT Title\_ratings\_pri\_key PRIMARY KEY (title\_id);

ALTER TABLE Title\_ratings

ADD CONSTRAINT Title\_ratings\_title\_id\_fkey FOREIGN KEY (title\_id) REFERENCES Titles(title\_id);

-- Issues with missing data in title.basics.tsv.gz, name.basics.tsv.gz, ...

-- Disable foreign key check lock

SET foreign\_key\_checks = 0;

-- Aliases has titles that do not exist in Titles, i.e., there are entries in

-- IMDb's title.akas.tsv.gz that are not present in title.basics.tsv.gz. The same

-- issue arises when setting the foreign key for the Alias\_attributes and

-- Alias\_types tables.

ALTER TABLE Aliases

ADD CONSTRAINT Aliases\_title\_id\_fkey FOREIGN KEY (title\_id) REFERENCES Titles(title\_id);

ALTER TABLE Alias\_attributes

ADD CONSTRAINT Alias\_attributes\_title\_id\_fkey FOREIGN KEY (title\_id) REFERENCES Titles(title\_id);

ALTER TABLE Alias\_types

ADD CONSTRAINT Alias\_types\_title\_id\_fkey FOREIGN KEY (title\_id) REFERENCES Titles(title\_id);

-- Ditto for Episode\_belongs\_to table.

ALTER TABLE Episode\_belongs\_to

ADD CONSTRAINT Episode\_belongs\_to\_show\_title\_id\_fkey FOREIGN KEY (parent\_tv\_show\_title\_id) REFERENCES Titles(title\_id);

ALTER TABLE Known\_for

ADD CONSTRAINT Known\_for\_title\_id\_fkey FOREIGN KEY (title\_id) REFERENCES Titles(title\_id);

ALTER TABLE Principals

ADD CONSTRAINT Principals\_name\_id\_fkey FOREIGN KEY (name\_id) REFERENCES Names\_(name\_id);

ALTER TABLE Principals

ADD CONSTRAINT Principals\_title\_id\_fkey FOREIGN KEY (title\_id) REFERENCES Titles(title\_id);

ALTER TABLE Had\_role

ADD CONSTRAINT Had\_role\_title\_id\_fkey FOREIGN KEY (title\_id) REFERENCES Titles(title\_id);

ALTER TABLE Had\_role

ADD CONSTRAINT Had\_role\_name\_id\_fkey FOREIGN KEY (name\_id) REFERENCES Names\_(name\_id);

ALTER TABLE Directors

ADD CONSTRAINT Directors\_name\_id\_fkey FOREIGN KEY (name\_id) REFERENCES Names\_(name\_id);

ALTER TABLE Writers

ADD CONSTRAINT Writers\_name\_id\_fkey FOREIGN KEY (name\_id) REFERENCES Names\_(name\_id);

-- Enable foreign key check lock

SET foreign\_key\_checks = 1;

-- Alias\_attributes

CREATE INDEX Alias\_attributes\_index ON Alias\_attributes(title\_id);

-- Alias\_types

CREATE INDEX Alias\_types\_index ON Alias\_types(title\_id);

-- Aliases

CREATE INDEX Aliases\_index ON Aliases(title\_id);

-- Directors

CREATE INDEX Directors\_title\_id\_index ON Directors(title\_id);

CREATE INDEX Directors\_name\_id\_index ON Directors(name\_id);

-- Episode\_belongs\_to

CREATE INDEX Episode\_belongs\_to\_ep\_title\_id\_index ON Episode\_belongs\_to(episode\_title\_id);

CREATE INDEX Episode\_belongs\_to\_show\_title\_id\_index ON Episode\_belongs\_to(parent\_tv\_show\_title\_id);

-- Had\_role

CREATE INDEX Had\_role\_title\_id\_index ON Had\_role(title\_id);

CREATE INDEX Had\_role\_name\_id\_index ON Had\_role(name\_id);

-- Known\_for

CREATE INDEX Known\_for\_index ON Known\_for(name\_id);

-- Name\_worked\_as

CREATE INDEX Name\_worked\_as\_index ON Name\_worked\_as(profession);

-- Names\_

CREATE INDEX Names\_index ON Names\_(name\_id);

-- Principals

CREATE INDEX Principals\_index ON Principals(title\_id);

-- Title\_genres

CREATE INDEX Title\_genres\_title\_id\_index ON Title\_genres(title\_id);

CREATE INDEX Title\_genres\_genre\_index ON Title\_genres(genre);

-- Title\_ratings

CREATE INDEX Title\_ratings\_index ON Title\_ratings(title\_id);

-- Titles

CREATE INDEX Titles\_index ON Titles(title\_id);

-- Writers

CREATE INDEX Writers\_title\_id\_index ON Writers(title\_id);

CREATE INDEX Writers\_name\_id\_index ON Writers(name\_id);